Direct, Prediction-based and Smoothing-based Particle Filter Algorithms

François Desbouvries [1] Boujemaa Ait-El-Fquih [2]

[1] TELECOM SudParis, CITI dpt., & CNRS UMR 5157, 9 rue Charles Fourier, 91011 Evry, France
E-mail: francois.desbouvries@it-sudparis.eu
[2]LSS & CNRS UMR 8506, Supélec, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette, France
E-mail: boujemaa.aitelfquih@lss.supelec.fr

Keywords: Kalman filters (KF), Sequential Monte Carlo (SMC), Particle Filtering (PF), Sequential Importance Sampling (SIS), Sampling Importance Resampling (SIR)

Abstract

We address the recursive computation of the a posteriori filtering probability density function (pdf) $p_{n|n}$ in a Hidden Markov Chain (HMC) model. We first observe that the classical path $p_{n-1|n-1} \rightarrow p_{n|n-1} \rightarrow p_{n|n}$ is not the only possible one that enables to compute $p_{n|n}$ recursively, and we explore the direct, prediction-based and smoothing-based recursive loops for computing $p_{n|n}$. We next propose a common methodology for computing these equations in practice. Since each path can be decomposed into a Bayesian step and a Markovian step, in the Gaussian case these two elementary operations are implemented by Gaussian transforms, and in the general case by elementary simulation techniques. By proceeding this way we obtain in parallel, for each filtering path, one set of Kalman filter (KF) equations and one generic sequential Monte Carlo (SMC) algorithm. Finally we get four KF (two of which are original), which themselves correspond to four generic SMC algorithms (two of which are original).

1 Introduction

Let $\mathbf{x}_n \in \mathbb{R}^m$ and $\mathbf{y}_n \in \mathbb{R}^p$ be respectively a hidden and observed process. Let $p(\mathbf{x}_n|\mathbf{y}_{0:n})$, say, denote the probability density function (pdf) (w.r.t. Lebesgue measure) of $\mathbf{x}_n$ given $\mathbf{y}_{0:n} = \{\mathbf{y}_i\}_{i=0}^n$ (other pdf or conditional pdfs of interest are defined similarly). Let $p(d\mathbf{x}) = p(\mathbf{x})d\mathbf{x}$ be the continuous measure with density $p(\mathbf{x})$. We assume that $\{\mathbf{x}_n, \mathbf{y}_n\}$ is an HMC : $p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = p(\mathbf{x}_0) \prod_{i=1}^n p(\mathbf{x}_i|\mathbf{x}_{i-1}) \prod_{i=0}^n p(\mathbf{y}_i|\mathbf{x}_i)$. We address the Bayesian filtering problem, which consists in (efficiently) computing $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ (at each time instant n), or at least an approximation of the measure $p(d\mathbf{x}_n|\mathbf{y}_{0:n})$ with density $p(\mathbf{x}_n|\mathbf{y}_{0:n})$.

In this paper we consider recursive solutions for computing $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. It is well known for instance that $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ can be computed from $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$ by

$$p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1}) p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}), \quad (1)$$

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n-1})}{\int p(\mathbf{y}_n|\mathbf{x}_n) p(d\mathbf{x}_n|\mathbf{y}_{0:n-1})}. \quad (2)$$

It remains to compute $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ in practice. In the linear Gaussian case, (1) (2) can be computed exactly. In the general case however, computing $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ is either difficult or impossible, and consequently many approximate techniques have been developed. Among them, PF (see e.g. (Doucet *et al.*(2000)) (Doucet *et al.* (2001)) (Cappé *et al.* (2005)) and references therein) are SMC methods which aim at propagating a discrete approximation of $p(d\mathbf{x}_n|\mathbf{y}_{0:n})$.

Our contribution in this paper is twofold. We first observe that (1)-(2), say, is one possible recursive formula for $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ which explicitely computes this pdf as the recursion $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) \rightarrow p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) \rightarrow p(\mathbf{x}_n|\mathbf{y}_{0:n})$ (or, in short, $p_{n-1|n-1} \rightarrow p_{n|n-1} \rightarrow p_{n|n}$). However following this path is not the only possible way to compute $p_{n|n}$ progressively, and indeed the first aim of this paper is to explore alternate recursions, even if $p_{n|n}$ is only obtained as a byproduct. For simplicity we will only consider those paths in which one time index is incremented at a time. The first alternative to consider is the path $p_{n-1|n-1} \rightarrow p_{n-1|n} \rightarrow p_{n|n}$. So both solutions compute $p_{n|n}$ recursively and differ only by the intermediate step which, for one of them, is the one step ahead predictive distribution $p_{n|n-1}$, and for the other the

one step backward smoothing distribution $p_{n-1|n}$. Now, these two densities can in turn be propagated from one time instant to the other via the two different paths which are obtained when moving one time index and next the other. This observation naturally yields six algorithms (four of which are distinct) for computing $p_{n|n}$ recursively; the two paths $p_{n-1|n-1} \rightarrow p_{n-1|n} \rightarrow p_{n|n}$ and $p_{n-1|n-1} \rightarrow p_{n|n-1} \rightarrow p_{n|n}$ are direct, i.e. $p_{n|n}$ is computed as the output of a recursive loop with input $p_{n-1|n-1}$; two other paths are prediction-based, i.e. $p_{n|n}$ is computed (indirectly) from the predictive distribution, but the recursion now acts on $p_{n|n-1}$; and finally two paths are smoothing-based.

We next address the practical computation of these four integral equations. We classically consider two cases : the Gaussian case, for which propagating densities amounts to propagating their parameters; and the general case, in which we resort to Monte Carlo (MC) approximations. However the novelty here relies in the systematic parallelization of the derivations. Since each path $p_{i|j} \rightarrow p_{i+1|j+1}$ consists of a Markovian step and a Bayesian step, we actually need a tool for implementing these two fundamental operations.

In the Gaussian case, the two steps are implemented by two elementary transformations among Gaussian variables. In the general case, the transformations among densities are replaced by the propagation of sets of points (approximately) sampled from these densities; the Markovian and Bayesian steps are then implemented by two elementary simulation techniques, and the optimal[1] conditional importance distribution (CID) for the filtering, prediction or smoothing problem naturally stems from one Bayes factorization of an associated fundamental transition pdf. Finally each path provides simultaneously one Kalman like and one SMC algorithm. Some of the KF solutions are well known but some are original; and similarly the SMC algorithms include well known SIR solutions (such as the bootstrap algorithm (Gordon *et al.*(1993))), but also SMC algorithms which are not SIS algorithms.

This paper is organized as follows. In §2 we briefly recall the PF methodology. In §3 we derive the direct filtering algorithms and address their implementation. Prediction- and smoothing-based algorithms are addressed respectively in §4 and 5, and we end the paper with some concluding remarks.

2 The PF methodology

2.1 The generic PF algorithm

Let us recall the principle of PF algorithms (see e.g. (Gordon *et al.*(1993)) (Doucet *et al.*(2000)) (Doucet *et al.* (2001)) (Cappé *et al.* (2005)) which are based on Importance Sampling (IS) techniques. Assume that at time $n-1$ we have a random discrete measure which approximates $p(d\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$:

$$p(d\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) \simeq \sum_{i=1}^{N} w_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}(d\mathbf{x}_{0:n-1}) \,, \tag{3}$$

where $\delta_x(.)$ is the Dirac mass at point x, the samples $\mathbf{x}_{0:n-1}^i$ are generated from an importance distribution $q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$, and the importance weight w_{n-1}^i associated to the i-th trajectory $\mathbf{x}_{0:n-1}^i$ is given by $w_{n-1}^i \propto p(\mathbf{x}_{0:n-1}^i|\mathbf{y}_{0:n-1}) / \; q(\mathbf{x}_{0:n-1}^i|\mathbf{y}_{0:n-1})$ with $\sum_{i=1}^N w_{n-1}^i = 1$. Let us see how to compute (3) recursively. To that end we start from

$$p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = \frac{p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1})}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})} p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}). \tag{4}$$

We first see how to update the trajectories. If we assume that the importance distribution factorizes as

$$q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n}) q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}), \tag{5}$$

i.e. that $q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$ is a marginal of $q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$, then for all $1 \leq i \leq N$, $[\mathbf{x}_{0:n}^i] = [\mathbf{x}_{0:n-1}^i, \mathbf{x}_n^i]$, in which $\mathbf{x}_n^i$ is sampled from the CID $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}^i, \mathbf{y}_{0:n})$. In other words, when stepping from time $n-1$ to time n, due to (5) we can keep the old trajectories $\{\mathbf{x}_{0:n-1}^i\}_{i=1}^N$, and we just need to extend each of them by sampling a new particle $\mathbf{x}_n^i$.

As for the weights w_n^i, we see from (4) that they can be computed recursively as

$$w_n^i \propto \underbrace{\frac{p(\mathbf{x}_n^i, \mathbf{y}_n|\mathbf{x}_{n-1}^i)}{q(\mathbf{x}_n^i|\mathbf{x}_{0:n-1}^i, \mathbf{y}_{0:n})}}_{\lambda_n^i} \times \underbrace{\frac{p(\mathbf{x}_{0:n-1}^i|\mathbf{y}_{0:n-1})}{q(\mathbf{x}_{0:n-1}^i|\mathbf{y}_{0:n-1})}}_{\propto w_{n-1}^i} . \tag{6}$$

[1] optimal in terms of the conditional variance of the importance weights.

Finally $\sum_{i=1}^N w_n^i \delta_{\mathbf{x}_{0:n}^i}(d\mathbf{x}_{0:n})$ approximates $p(d\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$, and thus $\sum_{i=1}^N w_n^i \delta_{\mathbf{x}_n^i}(d\mathbf{x}_n)$ is an MC approximation of $p(d\mathbf{x}_n|\mathbf{y}_{0:n})$.

2.2 Practical considerations

Now, SIS algorithms are well known to suffer from weights degeneracy. Two main rescues are available. First, it has proved important in the above generic algorithm to resample from $\sum_{i=1}^N w_n^i \delta_{\mathbf{x}_n^i}(d\mathbf{x}_n)$. Resampling can be performed either systematically or according to some strategy, and many variants are available, see for instance (Douc *et al.* (2005)). We shall not discuss this issue in this paper[2].

The second important point to take into account is to choose the CID carefully. Among other possibilities, two particular choices are of interest :

- sampling from the *a priori* transition kernel of the Markov chain $\mathbf{x}$ (i.e., choosing $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}^i, \mathbf{y}_{0:n}) = p(\mathbf{x}_n|\mathbf{x}_{0:n-1}^i) = p(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$, which was the original choice in the so-called Bootstrap filter (see (Gordon *et al.*(1993))) is popular because sampling from $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$ is often straightforward. Moreover, computing the incremental weight λ_n^i in (6) reduces to evaluating the conditional likelihood of the new observation given the updated particle position.

- However, this choice of the prior density can lead to poor performances, and it is often preferable (see (Doucet *et al.*(2000))) to sample the particles from the optimal CID

$$q^{opt}(\mathbf{x}_n|\mathbf{x}_{0:n-1}^i, \mathbf{y}_{0:n}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n), \tag{7}$$

  i.e. the distribution which minimizes the variance of the importance weights w_n^i, conditionally on the observations $\mathbf{y}_{0:n}$ and past samples $\mathbf{x}_{0:n-1}^i$. For this choice of the CID, $\lambda_n^i = p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$.

3 Direct filtering algorithms

In §2 we recalled the classical description of PF algorithms as SIS techniques. In this section we shall see that the bootstrap algorithm, and a reorganized version of the SIR algorithm with optimal CID, are indeed natural MC implementations of two two-steps direct filtering algorithms.

Let us start from (4) again. Due to the HMC assumption the fundamental transition pdf $p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1})$ can be factorized as

$$\begin{aligned} p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1}) &= p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{x}_{n-1}) \\ &= p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}) \qquad (8) \\ &= p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)p(\mathbf{y}_n|\mathbf{x}_{n-1}). \qquad (9) \end{aligned}$$

Injecting (8), (9) into (4) respectively gives (here $\mathcal{N}$ stands for numerator) :

$$p(\mathbf{x}_{n-1}, \mathbf{x}_n|\mathbf{y}_{0:n}) \overset{(8)}{=} \frac{p(\mathbf{y}_n|\mathbf{x}_n)\overbrace{[p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})]}^{p(\mathbf{x}_{n-1}, \mathbf{x}_n|\mathbf{y}_{0:n-1})}}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}) = \int \mathcal{N} d\mathbf{x}_{n-1} d\mathbf{x}_n} \tag{10}$$

$$\overset{(9)}{=} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)\underbrace{[\frac{p(\mathbf{y}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}) = \int \mathcal{N} d\mathbf{x}_{n-1}}]}_{p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})}. \tag{11}$$

Both formulas enable to compute $p_{n|n}$ from $p_{n-1|n-1}$; (10) (which coincides with (1)- (2) above) uses the path $p_{n-1|n-1} \to p_{n|n-1} \to p_{n|n}$, and (11) the path $p_{n-1|n-1} \to p_{n-1|n} \to p_{n|n}$. So in the first solution we predict $\mathbf{x}_n$, based on the same data $\mathbf{y}_{0:n-1}$, and then update $p_{n|n-1}$, thanks to $\mathbf{y}_n$; while in the second path updating comes first.

[2]In particular, the description of Algorithms 1 to 4 below always includes a systematic multinomial resampling step. This does not mean that we recommend this resampling scheme, but simply that these algorithms were derived from Rubin's Sampling Importance Resampling (SIR) mechanism (see §3.1.2, point 2). However, once a generic PF algorithm is obtained, other versions can be derived, which involve resampling only if some criterion is satisfied.

3.1 Practical computation

Two basic mechanisms are involved in equations (10) and (11) : a Markovian step which transforms some density $p(\mathbf{x}_1)$ into $p(\mathbf{x}_2) = \int p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1)d\mathbf{x}_1$; and a Bayesian step, which transforms $p(\mathbf{x})$ into $p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x})p(\mathbf{y}|\mathbf{x})}{\int p(\mathbf{x})p(\mathbf{y}|\mathbf{x})d\mathbf{x}}$. Let us see how to implement these two transformations. In the Gaussian case transforming one density into another amounts to transforming its parameters; while in the general case the problem becomes that of transforming points sampled from $p(\mathbf{x}_1)$ (resp. from $p(\mathbf{x})$) into points sampled (at least approximatively) from $p(\mathbf{x}_2)$ (resp. from $p(\mathbf{x}|\mathbf{y})$).

3.1.1 The Gaussian case

1. Let $p(\mathbf{x}_1) \sim \mathcal{N}(\hat{\mathbf{x}}_1, \mathbf{P}_1)$ and $p(\mathbf{x}_2|\mathbf{x}_1) \sim \mathcal{N}(\mathbf{A}\mathbf{x}_1 + \mathbf{b}, \mathbf{P}_{2|1})$. Then $p(\mathbf{x}_2) \sim \mathcal{N}\ (\mathbf{A}\widehat{\mathbf{x}}_1 + \mathbf{b},\ \mathbf{P}_{2|1} + \mathbf{A}\mathbf{P}_1\mathbf{A}^T)$.

2. Let $p(\mathbf{x}) \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$ and $p(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{P}_{\mathbf{y}|\mathbf{x}})$. Then

$$p(\mathbf{x}|\mathbf{y}) \sim \mathcal{N}(\widehat{\mathbf{x}} + \mathbf{P}_{\mathbf{x}}\mathbf{A}^T[\mathbf{P}_{\mathbf{y}|\mathbf{x}} + \mathbf{A}\mathbf{P}_{\mathbf{x}}\mathbf{A}^T]^{-1}(\mathbf{y} - \mathbf{A}\widehat{\mathbf{x}} - \mathbf{b}), \mathbf{P}_{\mathbf{x}} - \mathbf{P}_{\mathbf{x}}\mathbf{A}^T[\mathbf{P}_{\mathbf{y}|\mathbf{x}} + \mathbf{A}\mathbf{P}_{\mathbf{x}}\mathbf{A}^T]^{-1}\mathbf{A}\mathbf{P}_{\mathbf{x}}).$$

3.1.2 The general case

In all PF algorithms we sample (S) new particles, update some weights (W) and (possibly) resample (R) from a weighted discrete measure. These three SMC elementary operations S, W and R are indeed related to the following two well known generic simulation techniques, which in the context of this paper implement the Markovian and Bayesian operations.

1. (*Hierarchical sampling : S step*).

   Let $\mathbf{x}_1^i \sim p(\mathbf{x}_1)$ and let us sample $\mathbf{x}_2^i$ from $p(\mathbf{x}_2|\mathbf{x}_1^i)$. Then $(\mathbf{x}_1^i, \mathbf{x}_2^i)$ is a sample from $p(\mathbf{x}_1, \mathbf{x}_2)$ and thus $\mathbf{x}_2^i$ is a sample from $p(\mathbf{x}_2)$.

2. (*Rubin's SIR mechanism (Rubin (1988)) (Gelfand and Smith (1990)) (Smith and Gelfand (1992)) (Cappé et al. (2005),§9.2) : (W,R) steps*).

   Let $\tilde{p}(d\mathbf{x}) = \sum_{i=1}^N \frac{p(\mathbf{y}|\mathbf{x}^i)}{\sum_{i=1}^N p(\mathbf{y}|\mathbf{x}^i)}\delta_{\mathbf{x}^i}(d\mathbf{x})$ in which $\mathbf{x}^i$ are i.i.d. samples from $p(\mathbf{x})$. Conditionnally on $\{\mathbf{x}^i\}$, let $\{\tilde{\mathbf{x}}^i\}_{i=1}^I$ be i.i.d. samples from $\tilde{p}$; then $\{\tilde{\mathbf{x}}^i\}_{i=1}^I$ become i.i.d. samples from $p(\mathbf{x}|\mathbf{y})$ if $N \to \infty$.

3.2 Direct KF and SMC filters

We now use the tools of §3.1 to implement the exact formulas (10) and (11).

3.2.1 Implementing (10).

Let us first consider (10). The solution is well known in the context of linear Gaussian state-space systems. More precisely, let

$$\left\{ \begin{array}{rcl} \mathbf{x}_{n+1} & = & \mathbf{F}_n\mathbf{x}_n + \mathbf{G}_n\mathbf{u}_n \\ \mathbf{y}_n & = & \mathbf{H}_n\mathbf{x}_n + \mathbf{v}_n \end{array} \right. , \tag{12}$$

in which $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}}_0, \mathbf{P}_0)$, $\{\mathbf{u}_n\}$ and $\{\mathbf{v}_n\}$ are independent and mutually independent, $\mathbf{u}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_n)$ and $\mathbf{v}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_n)$. Then $p(\mathbf{x}_i|\mathbf{y}_{0:n}) \sim \mathcal{N}(\hat{\mathbf{x}}_{i|n}, \mathbf{P}_{i|n})$ for some mean vector $\hat{\mathbf{x}}_{i|n}$ and covariance matrix $\mathbf{P}_{i|n}$. Transforming $p_{n-1|n-1}$ into $p_{n|n-1}$ and $p_{n|n}$ via (10), reduces to transforming $(\hat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}_{n-1|n-1})$ into $(\hat{\mathbf{x}}_{n|n-1}, \mathbf{P}_{n|n-1})$ and $(\hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})$. This is done by the well known KF equations (Anderson & Moore (1979)), (Kailath *et al.* (2000)) :

$$\widehat{\mathbf{x}}_{n|n-1} = \mathbf{F}_{n-1}\widehat{\mathbf{x}}_{n-1|n-1}, \tag{13}$$

$$\mathbf{P}_{n|n-1} = \mathbf{F}_{n-1}\mathbf{P}_{n-1|n-1}\mathbf{F}_{n-1}^T + \mathbf{G}_{n-1}\mathbf{Q}_{n-1}\mathbf{G}_{n-1}^T, \tag{14}$$

$$\widehat{\mathbf{x}}_{n|n} = \widehat{\mathbf{x}}_{n|n-1} + \mathbf{P}_{n|n-1}\mathbf{H}_n^T(\mathbf{R}_n + \mathbf{H}_n\mathbf{P}_{n|n-1}\mathbf{H}_n^T)^{-1}(\mathbf{y}_n - \mathbf{H}_n\widehat{\mathbf{x}}_{n|n-1}), \tag{15}$$

$$\mathbf{P}_{n|n} = \mathbf{P}_{n|n-1} - \mathbf{P}_{n|n-1}\mathbf{H}_n^T(\mathbf{R}_n + \mathbf{H}_n\mathbf{P}_{n|n-1}\mathbf{H}_n^T)^{-1}\mathbf{H}_n\mathbf{P}_{n|n-1}. \tag{16}$$

Let us turn to MC approximations, beginning with the formula within brackets. We assume that at $n-1$ we have N samples from $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$. Using (§3.1.2, point 1) we get N samples $\tilde{\mathbf{x}}_n^i$ from $p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$ by drawing $\tilde{\mathbf{x}}_n^i$ from $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i) = p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_{0:n-1})$. Next using (§3.1.2, point 2), we get N (approximate) samples from $p(\mathbf{x}_n|\mathbf{y}_{0:n}) \propto p(\mathbf{y}_n|\mathbf{x}_n)\ p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$ by sampling from $\sum_{i=1}^N \frac{p(\mathbf{y}_n|\tilde{\mathbf{x}}_n^i)}{\sum_{i=1}^N p(\mathbf{y}_n|\tilde{\mathbf{x}}_n^i)} \delta_{\tilde{\mathbf{x}}_n^i}(d\mathbf{x}_n)$. We just described the sampling step S, followed by the updating step (W, R) of the Bootstrap algorithm

Algorithm 1. (= Bootstrap algorithm, Gordon *et al.*(1993))

Let $\widehat{p}(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N \frac{1}{N}\delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$ approximate $p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$.

S. For $1 \leq i \leq N$, sample $\tilde{\mathbf{x}}_n^i$ from $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$;

W. For $1 \leq i \leq N$, compute $w_n^i \propto p(\mathbf{y}_n|\tilde{\mathbf{x}}_n^i)$, $\sum w_n^i = 1$

R. For $1 \leq i \leq N$, sample $\mathbf{x}_n^i$ from $\sum_{i=1}^N w_n^i \delta_{\tilde{\mathbf{x}}_n^i}(d\mathbf{x}_n)$;

Then $\widehat{p}(d\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^N \frac{1}{N}\delta_{\mathbf{x}_n^i}(d\mathbf{x}_n)$ approximates $p(d\mathbf{x}_n|\mathbf{y}_{0:n})$.

3.2.2 Implementing (11).

Let us now address (11). Assuming (12) again, we get the following KF equations which, up to our best knowledge, are original :

$$\mathbf{K}_{n-1|n} = \mathbf{P}_{n-1|n-1}(\mathbf{H}_n^1)^T[\mathbf{H}_n^1\mathbf{P}_{n-1|n-1}(\mathbf{H}_n^1)^T + \mathbf{R}_n^1]^{-1} \quad (17)$$

$$\widehat{\mathbf{x}}_{n-1|n} = \widehat{\mathbf{x}}_{n-1|n-1} + \mathbf{K}_{n-1|n}(\mathbf{y}_n - \mathbf{H}_n^1\widehat{\mathbf{x}}_{n-1|n-1}) \quad (18)$$

$$\mathbf{P}_{n-1|n} = \mathbf{P}_{n-1|n-1} - \mathbf{K}_{n-1|n}[\mathbf{H}_n^1\mathbf{P}_{n-1|n-1}(\mathbf{H}_n^1)^T + \mathbf{R}_n^1]\mathbf{K}_{n-1|n}^T \quad (19)$$

$$\widehat{\mathbf{x}}_{n|n} = \mathbf{F}_{n-1}^1\widehat{\mathbf{x}}_{n-1|n} + \mathbf{b}_{n-1}^1 \quad (20)$$

$$\mathbf{P}_{n|n} = \mathbf{F}_{n-1}^1\mathbf{P}_{n-1|n}(\mathbf{F}_{n-1}^1)^T + \widetilde{\mathbf{Q}}_{n-1}^1 \quad (21)$$

with $\widetilde{\mathbf{Q}}_n = \mathbf{G}_n\mathbf{Q}_n\mathbf{G}_n^T$, $\mathbf{H}_n^1 = \mathbf{H}_n\mathbf{F}_{n-1}$, $\mathbf{R}_n^1 = \mathbf{H}_n\widetilde{\mathbf{Q}}_{n-1}\mathbf{H}_n^T + \mathbf{R}_n$, $\mathbf{F}_{n-1}^1 = \mathbf{F}_{n-1} - \widetilde{\mathbf{Q}}_{n-1}\mathbf{H}_n^T(\mathbf{R}_n^1)^{-1}\mathbf{H}_n^1$, $\mathbf{b}_{n-1}^1 = \widetilde{\mathbf{Q}}_{n-1}\mathbf{H}_n^T(\mathbf{R}_n^1)^{-1}\mathbf{y}_n$ and $\widetilde{\mathbf{Q}}_{n-1}^1 = \widetilde{\mathbf{Q}}_{n-1} - \widetilde{\mathbf{Q}}_{n-1}\mathbf{H}_n^T(\mathbf{R}_n^1)^{-1}\mathbf{H}_n\widetilde{\mathbf{Q}}_{n-1}$.

Let us now address the general case. Assume again that at $n-1$ we have N samples $\mathbf{x}_{n-1}^i$ from $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$. From the SIR mechanism described in (§3.1.2, point 2), we compute weights w_n^i proportional to $p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$; $\sum_{i=1}^N w_n^i\delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$ approximates $p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$, and (re)sampling from this distribution provides (approximate) samples $\{\tilde{\mathbf{x}}_{n-1}^i\}_{i=1}^N$ from $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$. Using (§3.1.2, point 1), we finally sample $\mathbf{x}_n^i$ from $p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n) = p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_{0:n})$. We obtained Algorithm 2, which coincides with (Cappé *et al.* (2005),Algorithm 8.1.1. p. 253) :

Algorithm 2.

Let $\widehat{p}(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N \frac{1}{N}\delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$ approximate $p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$.

W. For $1 \leq i \leq N$, compute $w_n^i \propto p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$, $\sum_{i=1}^N w_n^i = 1$;

R. For $1 \leq i \leq N$, sample $\tilde{\mathbf{x}}_{n-1}^i \sim \sum_{i=1}^N w_n^i\delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$;

S. For $1 \leq i \leq N$, sample $\mathbf{x}_n^i$ from $p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)$.

Then $\widehat{p}(d\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^N \frac{1}{N}\delta_{\mathbf{x}_n^i}(d\mathbf{x}_n) \simeq p(d\mathbf{x}_n|\ \mathbf{y}_{0:n})$.

Comparing with §2.2, we see that Algorithm 2 is a reordering of the SIR algorithm with optimal CID (and with systematic resampling) (Doucet *et al.*(2000)). However, these two algorithms are not simply related by a shift in time. More precisely, in (Doucet *et al.*(2000)) the successive steps are $S \to W \to R$ (or, equivalently, $W \to S \to R$: steps S and W commute since λ_n^i does not depend on the new particle $\mathbf{x}_n^i$), while the recursive loop of Algorithm 2 is made of the successive steps $W \to R \to S$. So both loops coincide in the absence of resampling. However if there is resampling, $\mathbf{x}_n^i$ is sampled from $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$ in Doucet *et al.*(2000), while in Algorithm 2 $\mathbf{x}_n^i \sim p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)$ in which $\tilde{\mathbf{x}}_{n-1}^i \sim \sum_{i=1}^N w_n^i\delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$; in others words, each old particle $\mathbf{x}_{n-1}^i$ is taken equally into account in Doucet *et al.*(2000), while only those with high weights do contribute to the updated trajectory in Algorithm 2. In section 3.3 below we now explain more deeply on the differences between both algorithms.

3.3 SIR with optimal CID : original or reorganized ?

3.3.1 Sampling before updating

The original algorithm is, by nature, a batch IS algorithm which has been rendered adaptive, but in which the trajectories, in the absence of resampling, tend to diverge (as time increases) from properly sampled trajectories (i.e. from trajectories sampled from the target density $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$).

More precisely, at fixed time $n-1$, we would like to compensate the missing pdf $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$ by multiple imputations from $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$, and next replace the corresponding expectations accordingly; since it is easier to sample from q, we indeed dispose of the weighted measure $\hat{p}(d\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}(d\mathbf{x}_{0:n-1})$, in which the N trajectories $\mathbf{x}_{0:n-1}^i$ are sampled from q and $w_{n-1}^i \propto \frac{p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})}{q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})}$.

Let us now address adaptivity. Due to condition (5) on the importance function q, this IS algorithm can be sequentialized easily : each old trajectory $\mathbf{x}_{0:n-1}^i$ is kept unaltered and is simply extended by one new particle $\mathbf{x}_n^i$; moreover the associated weight can be updated recursively. On the other hand, a drawback of this simple computational scheme is that in the absence of resampling, the trajectories do not fully take into account the data : the new datum $\mathbf{y}_n$ can only modify (at least if the optimum CID is used) the present and future values of the trajectories, but not the past ones. More precisely, assume that resampling has occured at time k, but not between k and n with $n > k$. From §3.1.2, point 2, $\mathbf{x}_{0:k}^i$ are (approximately) sampled from $p(\mathbf{x}_{0:k}|\mathbf{y}_{0:k})$[3], and trajectories $\mathbf{x}_{0:n}^i$, which should be sampled from

$$p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) \propto p(\mathbf{x}_{0:k}|\mathbf{y}_{0:k}) \prod_{j=k+1}^n p(\mathbf{x}_j|\mathbf{x}_{j-1}) p(\mathbf{y}_j|\mathbf{x}_j) \tag{22}$$

$$\propto p(\mathbf{x}_{0:k}|\mathbf{y}_{0:k}) \prod_{j=k+1}^n p(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{y}_j) p(\mathbf{y}_j|\mathbf{x}_{j-1}), \tag{23}$$

are indeed sampled either from

$$q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = p(\mathbf{x}_{0:k}|\mathbf{y}_{0:k}) \prod_{j=k+1}^n p(\mathbf{x}_j|\mathbf{x}_{j-1}) = p(\mathbf{x}_{0:n}|\mathbf{y}_{0:k}) \tag{24}$$

(if the bootstrap algorithm is used), or from

$$q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = p(\mathbf{x}_{0:k}|\mathbf{y}_{0:k}) \prod_{j=k+1}^n p(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{y}_j) \tag{25}$$

(if the SIR algorithm with optimal CID is used). We see that for $0 \leq j \leq n$, each particle $\mathbf{x}_j^i$ does not depend on the whole data $\mathbf{y}_{0:n}$, but only on $\mathbf{y}_{0:k}$ (if (24) is used) or on $\mathbf{y}_{0:\max(k,j)}$ (if (25) is used). So trajectories $\mathbf{x}_{0:n}^i$ sampled from (24) or (25) are all the more likely to diverge from trajectories sampled from (22) = (23) as n increases and departs from k. This discrepancy is (somehow) corrected by the weights, which take into account the new data $\mathbf{y}_{k+1:n}$.

3.3.2 Sampling after updating

By contrast, Algorithm 2 is not a SIS (or SIR) algorithm, in the sense that it is not a sequential version of an algorithm based on IS. Considered as a batch algorithm, at time $n-1$ $\hat{p}(d\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N \frac{1}{N} \delta_{\mathbf{x}_{0:n-1}^i}(d\mathbf{x}_{0:n-1})$, in which the N trajectories are (exactly) sampled from $q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) = p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$.

Let us now address the sequential version. At time n we need to sample from

$$\begin{aligned} q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) &= p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) \\ &= \underbrace{p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)}_{\substack{p(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{y}_n) \\ (\S 3.1.2,\ point\ 1)}} \times \underbrace{[\frac{p(\mathbf{y}_n|\mathbf{x}_{n-1})}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})} p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})]}_{\substack{p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n}) \\ (\S 3.1.2,\ point\ 2)}}. \end{aligned} \tag{26}$$

[3] though $\mathbf{x}_{0:k}^i$ are sampled from $p(\mathbf{x}_{0:k}|\mathbf{y}_{0:k})$ only if $N \to \infty$, we shall assume it is true, in particular in (24)-(25).

For this density the sufficient condition (5) is *not* satisfied (because of the multiplicative factor $\frac{p(\mathbf{y}_n|\mathbf{x}_{n-1})}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})}$ within the bracket), so the trajectories cannot be updated so easily. The presence of this factor means that we should transform trajectories sampled from $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$ into trajectories sampled from $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n})$. To that end we can use IS : from Rubin's sampling scheme (see §3.1.2, point 2), by updating the weights first we step from $p(d\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) \simeq \sum_{i=1}^N \frac{1}{N} \delta_{\mathbf{x}^i_{0:n-1}}(d\mathbf{x}_{0:n-1})$ to $p(d\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n})$ $\simeq \sum_{i=1}^N \frac{p(\mathbf{y}_n|\mathbf{x}^i_{n-1})}{\sum_{i=1}^N p(\mathbf{y}_n|\mathbf{x}^i_{n-1})} \delta_{\mathbf{x}^i_{0:n-1}}(d\mathbf{x}_{0:n-1})$, and next by resampling we get (approximate) trajectories from $p(d\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n})$. Comparing with SIR algorithms, instead of keeping all trajectories unaltered and then extending each of them by one new particle, in this scheme we reweight all the trajectories first and then reselect them according to their weights. Once reselected, the i^{th} trajectory is extended by a new particle $\mathbf{x}^i_n$ (by using (3.1.2, point 1)).

Finally in SIR algorithms the batch estimates are fundamentally based on IS, but due to condition (5) no Bayes step is needed when updating the trajectories, and so no further IS mechanism is introduced. In Algorithm 2 the batch estimates are not based on IS, but updating trajectories requires a Bayes step followed by a Markovian step (see (26)), so IS is introduced locally as a means to implement the Bayes mechanism.

4 Prediction-based Filtering algorithms

For the prediction problem (4) is replaced by

$$p(\mathbf{x}_{0:n+1}|\mathbf{y}_{0:n}) = \frac{p(\mathbf{x}_{n+1}, \mathbf{y}_n|\mathbf{x}_n)}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})} p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n-1}). \tag{27}$$

The role played by the elementary transition $p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{x}_{n-1})$ in direct filtering algorithms is now played by $p(\mathbf{x}_{n+1}, \mathbf{y}_n|\mathbf{x}_n)$ which, due to Bayes's rule, can be factorized as :

$$\begin{aligned} p(\mathbf{x}_{n+1}, \mathbf{y}_n|\mathbf{x}_n) &= p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_{n+1}|\mathbf{x}_n) && (28) \\ &= p(\mathbf{x}_{n+1}|\mathbf{x}_n) p(\mathbf{y}_n|\mathbf{x}_n). && (29) \end{aligned}$$

So the two factorizations coincide if we commute the factors. However we should not, because injecting (28) and (29) in (27) leads to the following equations :

$$p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{y}_{0:n}) \overset{(29)}{=} p(\mathbf{x}_{n+1}|\mathbf{x}_n) \frac{\overbrace{p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n-1})}^{p(\mathbf{x}_n|\mathbf{y}_{0:n})}}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}) = \int \mathcal{N} d\mathbf{x}_n} \tag{30}$$

$$\overset{(28)}{=} \frac{p(\mathbf{y}_n|\mathbf{x}_n) \overbrace{[p(\mathbf{x}_{n+1}|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n-1})]}^{p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{y}_{0:n-1})}}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}) = \int \mathcal{N} d\mathbf{x}_n d\mathbf{x}_{n+1}} \tag{31}$$

So, as above, (30) and (31) describe the two different ways of moving from $p_{n|n-1}$ to $p_{n+1|n}$ which are obtained when incrementing one time index and next the other. Let us now consider practical implementations of these formulas.

4.1 Updating first (e.g., implementing (30))

Equation (30) describes the path $p_{n|n-1} \rightarrow p_{n|n} \rightarrow p_{n+1|n}$ which, up to a shift in time, coincides with the path $p_{n-1|n-1} \rightarrow p_{n|n-1} \rightarrow p_{n|n}$. As a consequence, implementing (30) yields the KF and SMC algorithm described in §3.2.1. At this point, it is interesting to observe that the optimal prediction CID is the prior $p(\mathbf{x}_n|\mathbf{x}_{n-1})$, which means that the Bootstrap algorithm is "optimal" (from that point of view) for the particle prediction problem.

4.2 Propagating the state first (e.g., implementing (31))

Taking the marginals w.r.t. $\mathbf{x}_n$ and $\mathbf{x}_{n+1}$, we see that (31) is a compact way of describing both the path $p_{n|n-1} \rightarrow p_{n+1|n-1} \rightarrow p_{n+1|n}$ (which is the recursive loop of the algorithm), and the path $p_{n|n-1} \rightarrow p_{n|n}$ (which was already obtained in (2)).

Assuming (12) again yields a solution which is well known in the context of linear Gaussian state-space systems (Anderson & Moore (1979),eqs. (4.9), (4.10), (4.12), (5.6) and (5.11)), (Kailath *et al.* (2000),Thm 9.5.1)) :

$$\begin{aligned}
\mathbf{L}_n &= \mathbf{H}_n\mathbf{P}_{n|n-1}\mathbf{H}_n^T + \mathbf{R}_n, && (32)\\
\mathbf{K}_{n+1|n} &= \mathbf{F}_n\mathbf{P}_{n|n-1}\mathbf{H}_n^T\mathbf{L}_n^{-1}, && (33)\\
\widehat{\mathbf{x}}_{n+1|n} &= \underbrace{\mathbf{F}_n\widehat{\mathbf{x}}_{n|n-1}}_{\widehat{\mathbf{x}}_{n+1|n-1}} + \mathbf{K}_{n+1|n}(\mathbf{y}_n - \mathbf{H}_n\widehat{\mathbf{x}}_{n|n-1}), && (34)\\
\mathbf{P}_{n+1|n} &= \underbrace{\mathbf{F}_n\mathbf{P}_{n|n-1}\mathbf{F}_n^T + \mathbf{G}_n\mathbf{Q}_n\mathbf{G}_n^T}_{\mathbf{P}_{n+1|n-1}} - \mathbf{K}_{n+1|n}\mathbf{L}_n\mathbf{K}_{n+1|n}^T, && (35)\\
\widehat{\mathbf{x}}_{n|n} &= \widehat{\mathbf{x}}_{n|n-1} + \mathbf{P}_{n|n-1}\mathbf{H}_n^T\mathbf{L}_n^{-1}(\mathbf{y}_n - \mathbf{H}_n\widehat{\mathbf{x}}_{n|n-1}), && (36)\\
\mathbf{P}_{n|n} &= \mathbf{P}_{n|n-1} - \mathbf{P}_{n|n-1}\mathbf{H}_n^T\mathbf{L}_n^{-1}\mathbf{H}_n\mathbf{P}_{n|n-1}. && (37)
\end{aligned}$$

We now consider the general case. Using §3.1.2 again, a direct MC implementation of this algorithm is given by the following algorithm :

Algorithm 3. Let $p(d\mathbf{x}_n|\mathbf{y}_{0:n-1}) \simeq \sum_{i=1}^N \frac{1}{N}\delta_{\mathbf{x}_n^i}(d\mathbf{x}_n)$.

Prediction.

S. For $i = 1, \cdots, N$, sample $\mathbf{x}_{n+1}^i \sim p(\mathbf{x}_{n+1}|\mathbf{x}_n^i)$;

W. For $i = 1, \cdots, N$, compute $w_n^i \propto p(\mathbf{y}_n|\mathbf{x}_n^i)$, $\sum_{i=1}^N w_n^i = 1$;

R. Resample from $\sum_{i=1}^N w_n^i \delta_{\mathbf{x}_{n+1}^i}(d\mathbf{x}_{n+1})$;

Then $p(d\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) \simeq \sum_{i=1}^N \frac{1}{N}\delta_{\mathbf{x}_{n+1}^i}(d\mathbf{x}_{n+1})$.

Filtering.

$p(d\mathbf{x}_n|\mathbf{y}_{0:n}) \simeq \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_n^i}(d\mathbf{x}_n)$.

5 Smoothing-based filtering algorithms

In this final section we see how $p_{n|n}$ can be computed recursively via the propagation of a smoothing distribution. We start from

$$p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n+1}) = \frac{p(\mathbf{x}_n, \mathbf{y}_{n+1}|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n})}{p(\mathbf{y}_{n+1}|\mathbf{y}_{0:n})} p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n}). \quad (38)$$

From the HMC model one can check easily that $(\mathbf{x}_n, \mathbf{y}_{n+1})$ is a Markov chain. So factor $p(\mathbf{x}_n, \mathbf{y}_{n+1}|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n})$ in (38) reduces to $p(\mathbf{x}_n, \mathbf{y}_{n+1}|\mathbf{x}_{n-1}, \mathbf{y}_n)$, which itself can be factorized as

$$\begin{aligned}
p(\mathbf{x}_n, \mathbf{y}_{n+1}|\mathbf{x}_{n-1}, \mathbf{y}_n) &= \underbrace{p(\mathbf{y}_{n+1}|\mathbf{x}_{n-1}, \mathbf{x}_n, \mathbf{y}_n)}_{p(\mathbf{y}_{n+1}|\mathbf{x}_n)} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n) && (39)\\
&= \underbrace{p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n, \mathbf{y}_{n+1})}_{q_S^{opt}} p(\mathbf{y}_{n+1}|\mathbf{x}_{n-1}, \mathbf{y}_n). && (40)
\end{aligned}$$

Injecting (39) and (40) into (38), and taking the marginal, we get

$$\begin{aligned}
p(\mathbf{x}_{n-1}, \mathbf{x}_n|\mathbf{y}_{0:n+1}) &= \frac{p(\mathbf{y}_{n+1}|\mathbf{x}_n)\overbrace{[p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})]}^{p(\mathbf{x}_{n-1}, \mathbf{x}_n|\mathbf{y}_{0:n})}}{p(\mathbf{y}_{n+1}|\mathbf{y}_{0:n}) = \int \mathcal{N} d\mathbf{x}_{n-1} d\mathbf{x}_n} && (41)\\
&= p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n, \mathbf{y}_{n+1}) \underbrace{[\frac{p(\mathbf{y}_{n+1}|\mathbf{x}_{n-1}, \mathbf{y}_n)p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})}{p(\mathbf{y}_{n+1}|\mathbf{y}_{0:n}) = \int \mathcal{N} d\mathbf{x}_{n-1}}]}_{p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n+1})}, && (42)
\end{aligned}$$

i.e. the two recursions on $p_{n-1|n}$ which, as above, are obtained by incrementing only one index and then the other. Equation (41) describes the path $p_{n-1|n} \to p_{n|n} \to p_{n|n+1}$, while equation (42) describes the path $p_{n-1|n} \to p_{n-1|n+1} \to p_{n|n+1}$. Let us now address practical implementations of these formulas.

5.1 Propagating before updating (e.g., implementing (41))

Just as in section 4.1, we observe that up to a shift in time, the path $p_{n-1|n} \rightarrow p_{n|n} \rightarrow p_{n|n+1}$, coincides with the path $p_{n-1|n-1} \rightarrow p_{n-1|n} \rightarrow p_{n|n}$. The KF and SMC implementations of (41) are thus the algorithms described in §3.2.2.

5.2 Propagating after updating (e.g., implementing (42))

Let us now consider (42), which is the smoothing counterpart of the prediction equation (31) (the path $p_{n-1|n} \rightarrow p_{n-1|n+1} \rightarrow p_{n|n+1}$ can be considered as the mirror path of $p_{n|n-1} \rightarrow p_{n+1|n-1} \rightarrow p_{n+1|n}$, in which the time indices of the observed and hidden processes are inverted). In the Gaussian case we get the following KF equations, which up to our best knowledge are original :

$$\begin{aligned}
\mathbf{K}_{n-1|n+1} &= \mathbf{P}_{n-1|n}(\mathbf{H}^2_{n+1})^T[\mathbf{H}^2_{n+1}\mathbf{P}_{n-1|n}(\mathbf{H}^2_{n+1})^T + \mathbf{R}^2_{n+1}]^{-1} && (43)\\
\widehat{\mathbf{x}}_{n-1|n+1} &= \widehat{\mathbf{x}}_{n-1|n} + \mathbf{K}_{n-1|n+1}(\mathbf{y}_{n+1} - \mathbf{H}^2_{n+1}\widehat{\mathbf{x}}_{n-1|n} - \mathbf{h}^2_{n+1}) && (44)\\
\mathbf{P}_{n-1|n+1} &= \mathbf{P}_{n-1|n} - \mathbf{K}_{n-1|n+1}[\mathbf{H}^2_{n+1}\mathbf{P}_{n-1|n}(\mathbf{H}^2_{n+1})^T + \mathbf{R}^2_{n+1}]\mathbf{K}^T_{n-1|n+1} && (45)\\
\widehat{\mathbf{x}}_{n|n+1} &= \mathbf{F}^2_{n-1}\widehat{\mathbf{x}}_{n-1|n+1} + \mathbf{b}^2_{n-1} && (46)\\
\mathbf{P}_{n|n+1} &= \mathbf{F}^2_{n-1}\mathbf{P}_{n-1|n+1}(\mathbf{F}^2_{n-1})^T + \widetilde{\mathbf{Q}}^2_{n-1} && (47)
\end{aligned}$$

with $\mathbf{H}^2_{n+1} = \mathbf{H}^1_{n+1}\mathbf{F}^1_{n-1}$, $\mathbf{h}^2_{n+1} = \mathbf{H}^1_{n+1}\mathbf{b}^1_{n-1}$, $\mathbf{R}^2_{n+1} = \mathbf{H}^1_{n+1}\widetilde{\mathbf{Q}}^1_{n-1}\mathbf{H}^T_{n+1} + \mathbf{R}^1_{n+1}$, $\widetilde{\mathbf{K}}_{n-1} = \widetilde{\mathbf{Q}}^1_{n-1}\,(\mathbf{H}^1_{n+1})^T\,(\mathbf{R}^2_{n+1})^{-1}$, $\mathbf{F}^2_{n-1} = \mathbf{F}^1_{n-1}$-$\widetilde{\mathbf{K}}_{n-1}\,(\mathbf{H}^1_{n+1})^T\mathbf{F}^1_{n-1}$, $\mathbf{b}^2_{n-1} = \widetilde{\mathbf{K}}_{n-1}\,(\mathbf{y}_{n+1} - \mathbf{H}^1_{n+1}\mathbf{b}^1_{n-1}) + \mathbf{b}^1_{n-1}$, and $\widetilde{\mathbf{Q}}^2_{n-1} = \widetilde{\mathbf{Q}}^1_{n-1} - \widetilde{\mathbf{K}}_{n-1}\mathbf{R}^2_{n+1}\widetilde{\mathbf{K}}^T_{n-1}$.

On the other hand, by using §3.1.2 an SMC implementation of (42) is given as follows :

Algorithm 4. Let $p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n}) \simeq \sum_{i=1}^N \frac{1}{N}\,\delta_{\mathbf{x}^i_{n-1}}(d\mathbf{x}_{n-1})$.

Smoothing.

- For $i = 1, \cdots, N$, compute $w^i_{n+1} \propto p(\mathbf{y}_{n+1}|\mathbf{x}^i_{n-1}, \mathbf{y}_n)$, $\sum_{i=1}^N w^i_{n+1} = 1$;
- Resample from $\sum_{i=1}^N w^i_{n+1}\delta_{\mathbf{x}^i_{n-1}}(d\mathbf{x}_{n-1})$. We get N points $\tilde{\mathbf{x}}^i_{n-1}$, (approximately) distributed $\sim p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n+1})$;
- For $i = 1, \cdots, N$, sample $\mathbf{x}^i_n \sim p(\mathbf{x}_n|\tilde{\mathbf{x}}^i_{n-1}, \mathbf{y}_n, \mathbf{y}_{n+1})$;
- Then $p(d\mathbf{x}_n|\mathbf{y}_{0:n+1}) \simeq \sum_{i=1}^N \frac{1}{N}\delta_{\mathbf{x}^i_n}(d\mathbf{x}_n)$.

Filtering.

- For $i = 1, \cdots, N$, sample $\overline{\mathbf{x}}^i_n \sim p(\mathbf{x}_n|\mathbf{x}^i_{n-1}, \mathbf{y}_n)$;
- Then $p(d\mathbf{x}_n|\mathbf{y}_{0:n}) \simeq \sum_{i=1}^N \frac{1}{N}\delta_{\overline{\mathbf{x}}^i_n}(d\mathbf{x}_n)$.

6 Conclusion

We derived six integral formulas for computing $p_{n|n}$: two of them directly compute $p_{n|n}$ from $p_{n-1|n-1}$, two compute $p_{n|n}$ via the prediction recursive loop ($p_{n|n-1} \rightarrow p_{n+1|n}$), and two compute $p_{n|n}$ via the smoothing recursive loop ($p_{n-1|n} \rightarrow p_{n|n+1}$). Each of these six formulas transforms $p_{j|k}$ into $p_{j+1|k+1}$ (with $(j,k) \in \{(n-1,n-1),(n,n-1),(n-1,n)\}$) by first updating one index and then the other, and the filtering pdf of interest $p_{n|n}$ is computed either directly (i.e., within the recursion) or indirectly.

We next considered both a KF and an SMC implementation of these formulas; since each of them consists of a propagation step and an updating step, the two basic operations involved are some transformations among Gaussian variables, or, in the general case, hierarchical sampling and Rubin's SIR simulation technique. This naturally leads to six KF and SMC algorithms, two of which are direct, two are prediction-based and two are smoothing-based. Among each pair of algorithms, the optimal solution (- optimal in terms of the conditional variance of the weights) is obtained when the updating step precedes the propagation step. Among the six solutions some coincide : the Bootstrap algorithm is both a direct

and prediction-based PF algorithm, and is optimal for the prediction loop, but not for the filtering loop; the reorganized SIR algorithm with optimal CID is both a direct and smoothing-based PF algorithm, and is optimal for the filtering loop, but not for the smoothing loop. Finally the two remaining prediction- and smoothing-based algorithms mirror each other. A summary of these algorithms is given in Table 1.

Nature	Eq.	Loop	q^{opt}	KF	SMC
Prediction -based	(31)	$p_{n\|n-1} \rightarrow p_{n+1\|n-1} \rightarrow p_{n+1\|n}$ $\downarrow$ $p_{n\|n}$		(32)-(37)	Alg. 3
	(30)	$p_{n-1\|n-1} \rightarrow p_{n\|n-1} \rightarrow p_{n\|n}$	$p(\mathbf{x}_n\|\mathbf{x}_{n-1})$	(13)-(16)	Bootstrap (Alg. 1)
Direct	(10)	$p_{n-1\|n-1} \rightarrow p_{n\|n-1} \rightarrow p_{n\|n}$			
	(11)	$p_{n-1\|n-1} \rightarrow p_{n-1\|n} \rightarrow p_{n\|n}$	$p(\mathbf{x}_n\|\mathbf{x}_{n-1},\mathbf{y}_n)$	(17)-(21)	Reorg. SIR + q_F^{opt} (Alg. 2)
Smoothing -based	(41)	$p_{n-1\|n} \rightarrow p_{n\|n} \rightarrow p_{n\|n+1}$			
	(42)	$p_{n-1\|n} \rightarrow p_{n-1\|n+1} \rightarrow p_{n\|n+1}$ $\downarrow$ $p_{n\|n}$	$p(\mathbf{x}_n\|\mathbf{x}_{n-1}^i,\mathbf{y}_n,\mathbf{y}_{n+1})$	(43)-(43)	Alg. 4

Table 1: Direct, Prediction- and Smoothing-based PF algorithms

References

Doucet, A., Godsill, S. J. and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, Vol. 10, 197-208.

Doucet A., de Freitas N. and Gordon N. (eds.) (2001). *Sequential Monte Carlo methods in practice.* Springer Verlag

Gordon, N. J., Salmond, D. J. and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*. Vol. 140, 107-113

Cappé, 0., Moulines E. and Rydén T. (2005) *Inference in Hidden Markov Models.* Springer-Verlag

Douc, R., Cappé, O. and Moulines, E. (2005) Comparison of Resampling Schemes for Particle Filtering. *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Zagreb, Croatia, September

Rubin, D. B. (1988). Using the SIR algorithm to simulate posterior distributions *Bayesian Statistics III.*

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling based approaches to calculating marginal densities *Journal of the American Statistical Association*. Vol. 85, N. 410, pp. 398-409

Smith, A. F. M. and Gelfand, A. E. (1992). Bayesian statistics without tears : a sampling-resampling perspective *The American Statistician*. Vol. 46, N. 2, pp. 84-87

Anderson, B. D. O. and Moore, J. B. (1979) *Optimal Filtering.* Prentice Hall, Englewood Cliffs, New Jersey

Kailath, T., Sayed, A. H. and Hassibi, B. (2000) *Linear estimation.* Prentice Hall